Natural Language Processing CSE 628, Assignment 2

In this Assignment we needed to implement the Dependency parser as per the paper "A Fast and Accurate Dependency Parser using Neural Network" and do different Experiments with the approach in order to explore different setups and their effect on the accuracy.

Experiments :

Number of Hidden Layers :

As we increase the hidden layers backpropagation algorithm becomes less effective. Since we are providing one more layer of random values we should provide enough training data to make those random values to detect some pattern in the data. In my case adding one more hidden layer resulted in slightly decrease in accuracy. Here we used the hidden layer size to be 100 and cubic activation function.

Results:

UAS	LAS
37.3459098754	30.8764336798

Capturing Interactions :

In place of cube non-linearity which is given in the paper I tried with different activation functions below are the results.

Results:

Sigmoid :

UAS	LAS
60.57986544	50.875335799

Results:

tanh

UAS	LAS
61.47518508	53.87242316

Results:

ReLU

UAS	LAS
60.7174016	52.82797816

Results:

Using cube non-linearity, in separate hidden layers

UAS	LAS
54.1269555372459	50.12839855677159

Word Embedding:

For word Embeddings I tried to generate the embeddings with cross entropy as well as NCE, below are the results.

Results:

Cross Entropy :

UAS	LAS
68.60931774559414	61.12869855672159

NCE:

UAS	LAS
66.09313565659414	60.1267855672159

Effect of Fixing words, POS, Dep. Embeddings:

This time I tried to fix the the embeddings and not allowing the embeddings to be modified via backpropogation. Below are the result.

Results:

UAS	LAS
23.7174016	20.82797816

Gradient Clipping:

When we are back propagating in the naural networks few gradients in the front layers can get too small because they are getting multiplied by small numbers. Similarly there can be a problem of exploding gradient too where gradients are getting bigger due to the multiplication by the number greater then one. Gradient clipping helps to keep the gradient values with in the bounds.

When I tried to remove the gradient clipping the loss was coming out to be nan after 3-4 iterations because gradient started exploding because of continuous multiplication of numbers greater then 1.

Best Configuration:

Results:

UAS	LAS
68.60931774559414	61.12869855672159

In my best Configuration I used

1. Cubic activation function with 1 hidden layer
2. Cross Entropy Word embedding
3. Without fixing the embedding allowing them to be updated while back propogation
4. Other configs looks like this

5.
   ```
   max_iter = 1001
   batch_size = 100
   hidden_size = 200
   embedding_size = 50
   ```

```
learning_rate = 0.1
display_step = 1
validation_step = 100
n_Tokens = 48
lam = 1e-8
```